



Module Specification

Operating Systems and Architecture

Version: 2026-27, v2.0, Approved

Contents

Module Specification	1
Part 1: Information	2
Part 2: Description	2
Part 3: Teaching and learning methods	3
Part 4: Assessment.....	4
Part 5: Contributes towards	6

Part 1: Information

Module title: Operating Systems and Architecture

Module code: UFCFCU-30-1

Level: Level 4

For implementation from: 2026-27

UWE credit rating: 30

ECTS credit rating: 15

College: College of Arts, Technology and Environment

School: CATE School of Computing and Creative Technologies

Partner institutions: Gloucestershire College

Field: Computer Science and Creative Technologies

Module type: Module

Pre-requisites: None

Excluded combinations: None

Co-requisites: None

Continuing professional development: No

Professional, statutory or regulatory body requirements: None

Part 2: Description

Overview: This module introduces students to the foundations of computer systems architecture together with the integrated hardware and software components and subsystems that enable and allow data to be input, processed and output. Low level programming is used to illustrate the operation of the components. The module then explores the concepts of operating systems, virtualisation, hardware management and file systems.

Students will learn the basic concepts of how data is represented in digital systems, logic and storage components, machine organisation and assembler programming. They will then build on this knowledge to see how levels of abstraction are introduced via operating systems and virtualisation to produce usable systems.

Features: Not applicable

Educational aims: This module contributes to the provision of underpinning technical understanding.

Outline syllabus: Topics may include, but are not limited to:

Classical computer architectures
Virtualised architectures
Digital logic, static and dynamic digital systems
Machine level representation of data
Assembly level machine organisation;
Memory system organisation and architecture
Interfacing and communication
Operating System principles
Concurrency and synchronisation
Scheduling and dispatch
Memory management
File systems
I/O system

Part 3: Teaching and learning methods

Teaching and learning methods: Lecture sessions cover the technical knowledge required. Designated practical work is included to ensure that apprentices have absorbed and understood the key principles involved.

This module will be based on ensuring that student's practical skills and knowledge gained in the block release sessions are carried into the workplace to inform their employment and generation of evidence of competency.

Module Learning outcomes: On successful completion of this module students will achieve the following learning outcomes.

MO1 Build test and debug a digital system to a specification.

MO2 Describe and explain computer architecture, digital logic, and machine-level representation of data.

MO3 Describe and explain the relationships between hardware components and the subsystems and the purposes and implementation of operating systems.

MO4 Implement simple programs in assembler language.

Hours to be allocated: 300

Contact hours:

Independent study/self-guided study = 135 hours

Placement = 75 hours

Face-to-face learning = 90 hours

Reading list: The reading list for this module can be accessed at [readinglists.uwe.ac.uk](https://uwe.rl.talis.com/index.html) via the following link <https://uwe.rl.talis.com/index.html>

Part 4: Assessment

Assessment strategy: Students grasp of fundamental operating systems programming is tested in a timed programming task. The emphasis is very much on being able to understand assembler code, rather than on demonstrating great fluency in writing the code. The students are therefore given an assembler program to analyse and describe its function. They must use it to explain the relationship between hardware components and programs.

In a second assessment task, the students will report on a practical task that they have carried out. During classroom sessions, the students will build a system consisting of a hypervisor, virtual client and virtual server. They will establish a client/server system and test functionality. They will write a report illustrating how their system has met the design requirement how it

operates with reference to detailed operating system functions.

The students report on their work in order to demonstrate their design thinking. The emphasis is on design in order to help them avoid a "build it and then fix it" mentality.

Assessment tasks:**Examination (First Sit)**

Description: 2 hour unseen exam

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3, MO4

Report (First Sit)

Description: 3000 word project report.

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO3

Examination (Resit)

Description: 2 hour unseen exam

Weighting: 50 %

Final assessment: No

Group work: No

Learning outcomes tested: MO2, MO3, MO4

Report (Resit)

Description: 3000 word project report.

Weighting: 50 %

Final assessment: Yes

Group work: No

Learning outcomes tested: MO1, MO3

Part 5: Contributes towards

This module contributes towards the following programmes of study:

Cyber Security Technical Professional {Apprenticeship-GLOSCOLL} [GlosColl] BSc
(Hons) 2026-27